Subject: Widget Objects
Posted by hamillnumerics on Fri, 25 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

An object-oriented approach to IDL widgets works nicely. Readers of this
newsgroup who'd like to see what I'm talking about are invited to check
out ...

http://hamillnumerics.home.mindspring.com/hamillnumerics/idl
/javaconcepts/widgetobject/widgetobjects1.html

Has someone already done this?

--
Jim Hamill
Hamill Numerics
in beautiful East Tennessee

Subject: Re: Widget Objects
Posted by John-David T. Smith on Mon, 28 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Jim Hamill (hamillnumerics@mindspring.com) writes:
>
>> An object-oriented approach to IDL widgets works nicely. Readers of this
>> newsgroup who'd like to see what I'm talking about are invited to check
>> out ...
>>
>> Has someone already done this?
>
> Uh, no. Never occurred to me.
>
> Cheers,
>
> David
>
> P.S. Excuse me while I go bang my head against the wall here. :-(

I think David means to say... go search on DejaNews.com for threads such as
"Object Widget", "Objet" which date back up to a year or more. There's been
quite a bit of talk on this paradigm. I *still* think it would be useful to
bang heads and come up with a commonly agreed-upon superclass for this (and then
convincing RSI to include it), which provides high level message passing and
simple, but powerful life-cycle control.

Of course, that's for another day...

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: widget objects
Posted by MKatz843 on Wed, 31 Mar 2004 20:23:52 GMT
View Forum Message <> Reply to Message

Regarding object widgets, a fun topic for discussion if there ever were.
I was rummaging through IDL's libraries and came across this object class. (Your path may be somewhat different)

/Applications/idl_6.0/lib/utilities/idlexwidget__define.pro

Here's the top of the file's header. This would appear to be an RSI object cw_widget prototype. I just through it might be of interest to the group to know that this is there.
-M. Katz

```
;$Id: idlexwidget__define.pro,v 1.7 2003/02/03 18:14:12 scottm Exp $
;
;  Copyright (c) 1997-2003, Research Systems, Inc. All rights reserved.
;       Unauthorized reproduction prohibited.
;
; NAME:
;   IDLexWidget
;
; PURPOSE:
;   Provide a base class for defining objects that function
;       as IDL compound widgets.
;
; CATEGORY:
;   Widgets.  Object Oriented Programming.
;
; CREATION:
;   It is intended that subclasses of class IDLexWidget, rather than
;   class "IDLexWidget" itself, get instantiated.  It is possible to
;   create a "IDLexWidget" object directly...
;
```

```
;      oWidget = OBJ_NEW('IDLexWidget', wParent)
;
;   ... but such objects would not have any buttons, sliders, etc.
;   in them, and would have a propensity to throw errors.
;
```

Subject: Re: widget objects
Posted by JD Smith on Wed, 31 Mar 2004 20:48:04 GMT
View Forum Message <> Reply to Message

On Tue, 30 Mar 2004 15:56:25 -0500, Michael A. Miller wrote:

> I've got some questions regarding objects and widgets and am
> hoping that someone here can help point me toward a robust
> design. Here's the situation...
>
> I have a collection of IDL objects that include GUI components.
> I use these in two ways: by themselves in their own top level
> windows and swallowed in widget_bases of other GUIS, like a
> compound widget. This works pretty well for me, but has some
> limitations. For example, I have a color map object that allows
> me to select a color map and adjust the limits. It inherits from
> a publisher class so other objects that might use it can
> subscribe. When the color map is changed, it tells its
> subscribers. Typically a subscriber will redraw something using
> the color map when it is notified of a change.
>
> Now I've decided that in some cases, it's is ok to have a color
> map GUI on the screen (as part of a main window or in its own
> window) all the time. In other cases, I'd really like to put the
> color map in its own window which can be drawn and withdrawn at
> will, like with tk's withdrawn window state. I don't believe
> that this is possible with IDL, so instead I thought I'd separate
> the color map object from the gui so that an instance can create
> and activate the gui and destroy it at will (at the programmers
> will, that is) without destroying the entire object.
>
> This sounds fairly straight forward, but I'm confident that I can
> make it complex enough to be cumbersome. I wonder if any of yous
> idlers have given this sort of thing some thought and have ideas
> that you'd be willing to share or just discuss in general. The
> part in particular that I'm wondering about is the capability of
> putting a gui component in it's own top level base or in another
> window. I've implemented this with a widget object class which
> is appended below. Using it just seems a bit cumbersome to me
> and I can't quiet put my finger on why.

I do this without any difficulty. I usually have a separate method call which actually contains the widget definition code and (optionally) XManager call. I call it when necessary, and provide a means to test if the GUI is running (XRegistered is convenient, as is widget_info(/VALID_ID)). To have it function either standalone or as a compound widget, pass an optional parent widget ID. If the parent ID is passed and is valid, build all widgets beneath it, and don't call XManager (since the parent application will do that). If it's not passed, make your own TLB (perhaps MODAL) and proceed the same way, calling XManager this time to manage your own events. In either case, the events will pass through your own event handler (which, since you can't rely on XManager to set for you, you should do using WIDGET_CONTROL on the necessary top-level widget). The widget object doesn't know whether it's running *inside* another widget or as a stand-alone. Nor does it care whether it actually has any widgets realized and showing. The only potential difficulty appears if you want to pass events "up" from your widget-object-as-compound-widget to be handled directly in the parent app's event handler. You can do away with this difficulty is you simply stick to your publish/subscribe based communication in all cases, and swallow all events (keeping them internal to your object's widget hierarchy). This methodology will pay off in the long run, since then you can reuse the same widget object in a variety of ways, and its communication pathways remain flexible.

To have, e.g., a colormap widget set appear and disappear within another base, you can just wrap the widget building code in widget_control, self.parent,UPDATE=0, widget_control, self.parent,UPDATE=1 and destroy it when it's not popped up. Since the object lives through this process, there is no problem maintaining state: it simply appears that the widget disappears and reappears on command. Except for the heaviest of widgets, this is acceptably fast.

JD

Subject: Re: widget objects
Posted by btt on Wed, 31 Mar 2004 20:53:22 GMT
View Forum Message <> Reply to Message

Michael A. Miller wrote:

> I've got some questions regarding objects and widgets and am
> hoping that someone here can help point me toward a robust
> design. Here's the situation...
>

I'm not sure if it would be helpful to you now, since you appear to be

so far along, but you should check out Martin Schultz's object library.
In particular, take a peek at MGS_BaseGUI. He built the behavior you describe into this object.

http://www.mpimet.mpg.de/en/misc/software/idl/

Ben

Subject: Re: widget objects
Posted by Paul Sorenson on Mon, 05 Apr 2004 21:18:12 GMT
View Forum Message <> Reply to Message

My visio diagram includes this class. The diagram is at:
http://www.paulsorenson.com/underthehood.html

-Paul

"M. Katz" <MKatz843@onebox.com> wrote in message
news:4a097d6a.0403311223.462cd994@posting.google.com...
> Regarding object widgets, a fun topic for discussion if there ever
> were.
> I was rummaging through IDL's libraries and came across this object
> class. (Your path may be somewhat different)
>
> /Applications/idl_6.0/lib/utilities/idlexwidget__define.pro
>
> Here's the top of the file's header. This would appear to be an RSI
> object cw_widget prototype. I just through it might be of interest to
> the group to know that this is there.
> -M. Katz
>
> ;$Id: idlexwidget__define.pro,v 1.7 2003/02/03 18:14:12 scottm Exp $
> ;
> ; Copyright (c) 1997-2003, Research Systems, Inc. All rights
> reserved.
> ; Unauthorized reproduction prohibited.
> ;
> ; NAME:
> ; IDLexWidget
> ;
> ; PURPOSE:
> ; Provide a base class for defining objects that function
> ; as IDL compound widgets.
> ;
> ; CATEGORY:
> ; Widgets. Object Oriented Programming.
> ;

```
> ; CREATION:
> ;   It is intended that subclasses of class IDLexWidget, rather than
> ;   class "IDLexWidget" itself, get instantiated.  It is possible to
> ;   create a "IDLexWidget" object directly...
> ;
> ;      oWidget = OBJ_NEW('IDLexWidget', wParent)
> ;
> ;   ... but such objects would not have any buttons, sliders, etc.
> ;   in them, and would have a propensity to throw errors.
> ;
```
